Aquaristik

Aufgaben

Für die Zucht von Süßwasserlebewesen (Zierfische und Wasserpflanzen) sowie den Handel von Artikeln rund um das Thema Aquaristik sollen eine Datenbank sowie eine Applikation zum Management der Zuchtanlage entwickelt werden.

1 Entwicklung einer Datenbank für den Aquaristikhandel
Ein erstes Entity-Relationship-Modell (ERM) ist in Material 1 dargestellt.

1.1 Beschreiben Sie die in dem ERM auftretenden Notationselemente und erklären Sie deren Bedeutung jeweils anhand eines Beispiels aus dem vorliegenden Modell.

(6 BE)

1.2 Überführen Sie das ERM in das relationale Modell in der 3. Normalform und begründen Sie Ihre Entscheidungen.

Hinweise: Es ist darauf zu achten, dass möglichst keine null-Werte in den Datenfeldern auftreten. Alle Relationen sind in der Schreibweise `Relation(PK, Attribut, ..., FK#)` anzugeben.

(9 BE)

1.3 Einige Daten der Datenbank sollen ergänzt, geändert bzw. ausgewertet werden.

1.3.1 Der neue Kunde Hans Hecht aus 89075 Ulm, Donautor 1, Telefon 0177 1234567 erhält die Kundennummer 7654 und bestellt am 02.05.2022 einen Aquariumbelüfter (`artNr` 277). Entwickeln Sie die SQL-Anweisungen, um die erforderlichen Datensätze einzufügen und den Bestand des Artikels zu aktualisieren.

Hinweis: Die Bestellnummer einer Bestellung wird automatisch vergeben (`AUTO_INCREMENT`).

(7 BE)

1.3.2 Formulieren Sie eine SQL-Anweisung, die die Bestellnummern und Kundendaten der Bestellungen, die noch nicht geliefert wurden und in den PLZ-Bereich 12 versendet werden sollen, ausgibt.

(3 BE)

1.3.3 Die Gesamtumsätze von Fischen der Bestellungen mit dem Bestelldatum im 1. Quartal 2022 sollen nach Herkunft gruppiert ausgegeben werden.
Implementieren Sie eine entsprechende SQL-Anweisung.

(5 BE)

1.4 Im Aquaristikhandel werden Daten zum Versand der Artikel einer Bestellung erfasst, die zurzeit noch in Form einer Tabelle (Material 2) festgehalten werden.
Entwickeln und zeichnen Sie die Erweiterung des ERM in Material 1 auf Grundlage der Tabelle in Material 2.

Hinweis: Es sind nur die Erweiterungen und Veränderungen zum ursprünglichen Modell zu berücksichtigen.

(10 BE)

2 Objektorientierte Entwicklung eines Zuchtmanagementsystems
Ein erstes UML-Klassendiagramm finden Sie in Material 3.

2.1 Beschreiben Sie das Prinzip der Datenkapselung unter Nennung der verschiedenen Zugriffsmodifizierer aus dem Klassendiagramm und erläutern Sie dieses Prinzip am Beispiel des Containers `lebewesen` der Klasse `Aquarium`.

(4 BE)

2.2 In Material 4 ist beispielhaft der Besatz zweier Aquarien aufgeführt. Entwickeln und zeichnen Sie auf Grundlage von Material 4, unter Berücksichtigung der dort aufgeführten Hinweise sowie des Klassendiagramms, ein Objektdiagramm in UML-Notation.

(8 BE)

2.3 Überführen Sie die Klassen `Aquarium`, `Lebewesen` und `Fischart` in Anweisungen einer objektorientierten Programmiersprache und implementieren Sie die dargestellten Methoden.

Hinweise: Die Methode `ermittleHerkunft()` der Klasse `Aquarium` gibt die Herkunft des ersten Lebewesens oder einen leeren String bei einem leeren Aquarium zurück. Die Methode `pruefeFreiesVolumen()` gibt `true` zurück, wenn das Volumen des Aquariums für die Gruppe der übergebenen Fischart ausreicht, ansonsten `false`. Die Methode `besetzeAquarium()` fügt die übergebene Art dem Container `lebewesen` hinzu. Bei Fischarten ist das nur möglich, wenn das freie Volumen des Aquariums ausreichend ist. Sie können davon ausgehen, dass der Methode `besetzeAquarium()` nur verträgliche Fischarten übergeben werden. Die Dokumentation der Klasse `List` ist in Material 5 zu finden.

(16 BE)

2.4 Die Methode `findeAquarium()` der Klasse `Zuchtmanagement` gibt ein Aquarium für die übergebene Fischart zurück, wobei folgende Festlegungen zu berücksichtigen sind:

- Es ist immer mindestens ein leeres Aquarium vorhanden.
- Ein Aquarium muss über ausreichend freien Platz für die Gruppengröße der Fischart verfügen.
- In einem Aquarium haben alle Lebewesen dieselbe Herkunft.
- Die Fischarten in einem Aquarium müssen untereinander verträglich sein. Eine Fischart, die mit keiner anderen verträglich ist, wird als einzige in einem Aquarium gehalten.
- Ein Aquarium wird nur einmal mit derselben Fischart besetzt.
- Der Besatz einer Fischart in ein bereits mit Lebewesen gleicher Herkunft belegtes Aquarium ist bei ausreichendem Platz vorzuziehen. Wenn das nicht möglich ist, wird das erste leere Aquarium des Containers `aquarien` zurückgegeben.

Entwickeln und zeichnen Sie ein Struktogramm für die Methode.

(12 BE)

2.5 Die Steuerung von Temperatur und Licht soll automatisch erfolgen, wobei für jedes Aquarium aufgrund der jeweiligen Lebensgemeinschaft spezifische Werte einzuhalten sind. Jedes Aquarium wird mit einem USB[1]-Interface der Firma Velleman ausgestattet. In einem ersten Schritt soll die Temperaturregelung umgesetzt werden. Die entsprechende Erweiterung des UML-Klassendiagramms ist in Material 6 zu finden. Die Temperaturerfassung erfolgt über einen entsprechenden Sensor, als Aktor dient ein Heizstab. Zusätzlich wird eine Alarmleuchte eingesetzt. Die Belegung der Ports des USB-Interfaces ist in Material 7 schematisch dargestellt.

2.5.1 Die Klasse `Steuerung` (Material 6) überwacht und regelt die Temperatur eines Aquariums.

- Bei der Erzeugung eines Steuerungsobjekts für das übergebene Aquarium auf dem angegebenen Device wird die optimale Temperatur (Attribut `optimaleTemp`) für die Lebensgemeinschaft des Aquariums anhand der Methode `ermittleOptimaleTemp()` initialisiert.
- Die Methode `ermittleAktuelleTemp()` gibt einen über 20 eingelesene Messwerte des Temperatursensors gemittelten und in Grad Celsius umgewandelten Wert zurück, wobei zwischen den einzelnen Messungen 10 Millisekunden Wartezeit liegen.
- Die Methode `run()` der Klasse `Steuerung` schaltet den Heizstab ein und prüft im Minutentakt das Erreichen der optimalen Temperatur. Ist diese erreicht, wird der Heizstab entsprechend ausgeschaltet.
- Zur daraufhin beginnenden Überwachung wird die aktuelle Temperatur ermittelt und mit der optimalen Temperatur verglichen. Um ständige Schaltvorgänge zu verhindern, ist eine Abweichung von ± 0,5 Grad Celsius zu berücksichtigen. Ein Messintervall beträgt in der Regel 5 Minuten.
- Wenn eine Messung unter oder über dem Sollbereich liegt, wird der Heizstab entsprechend geschaltet, das Ereignis gezählt und das Messintervall auf 1 Minute heruntergesetzt.
- Wenn der Sollbereich nach 5 Minuten nicht wiederhergestellt ist, wird die Alarmleuchte eingeschaltet.

Implementieren Sie Klasse `Steuerung` mit dem Konstruktor und den Methoden `run()` sowie `ermittleAktuelleTemp()`.

Hinweis: Die Dokumentation der Klasse `VUSB` ist in Material 5 zu finden.

(12 BE)

2.5.2 Die Steuerung soll um die Lichtregelung mit folgenden Anforderungen erweitert werden:

- Eine dimmbare LED-Beleuchtung für ein Aquarium zeichnet sich durch ihr Modell und die maximale Leistung in Lumen[2] aus. Die optimale Lichtmenge, die Farbtemperatur[3] und die Tageslänge (Dauer der Beleuchtung pro Tag) werden entsprechend der Herkunft der Lebensgemeinschaft festgelegt.
- Die Lichtsteuerung soll durch entsprechendes Dimmen der Leuchte einen natürlichen Tagesablauf simulieren (Sonnenaufgang, Sonnenuntergang, Mondphasen usw.).
- Die Steuerung von Temperatur und Licht müssen gleichzeitig ablaufen.

Entwickeln Sie auf Grundlage von Material 6 ein UML-Klassendiagramm mit Attributen, Konstruktoren, Methoden und Assoziationen, das die genannten Anforderungen zusätzlich umsetzt.

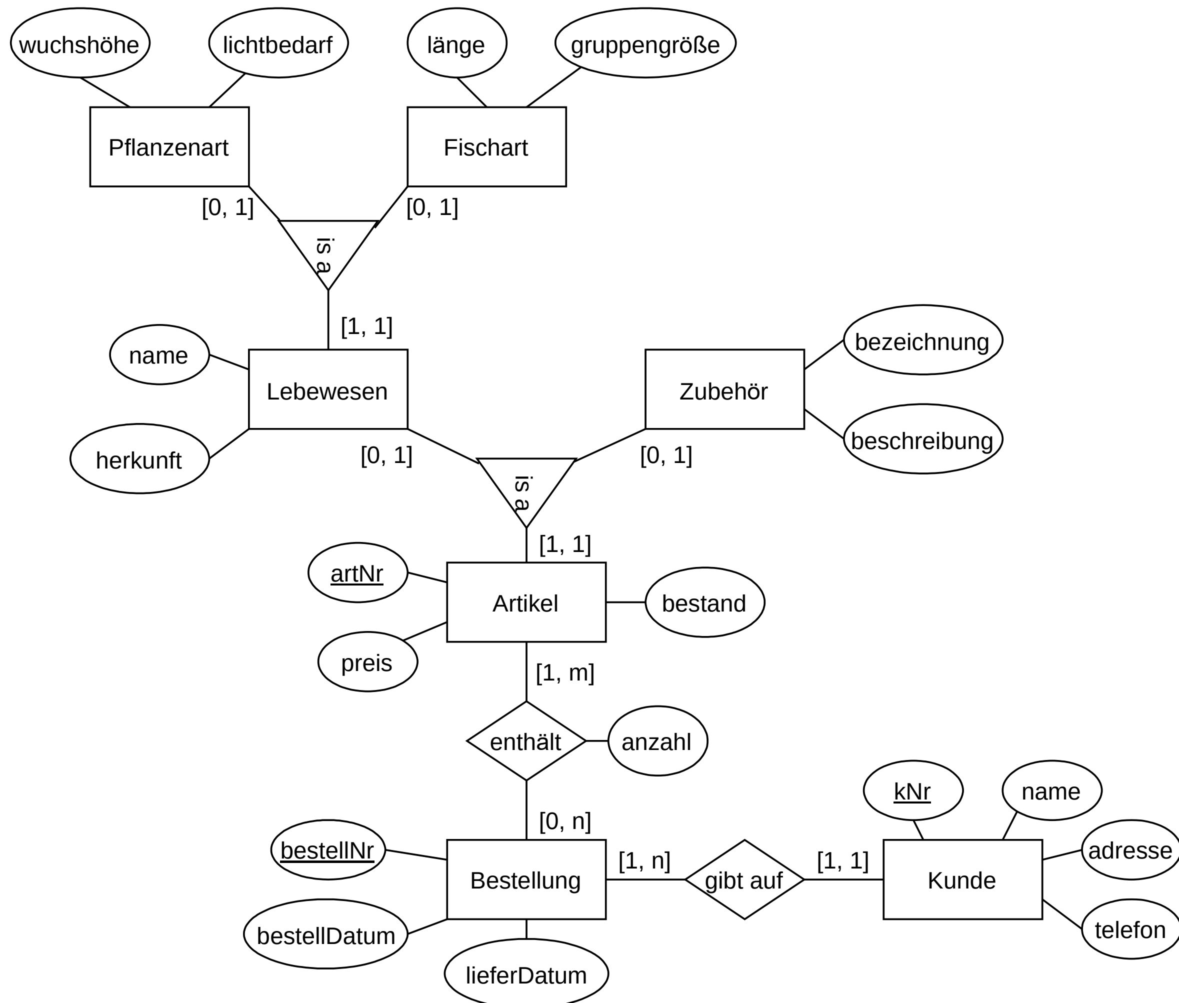
Hinweis: Set- und get-Methoden müssen nicht dargestellt werden.

(8 BE)

[1] USB: Universal Serial Bus
[2] Das Lumen ist die Einheit des Lichtstroms. Es gibt an, wie viel Licht eine Lichtquelle pro Zeiteinheit insgesamt abgibt.
[3] Die Farbtemperatur (Einheit Kelvin) ist ein Maß, um den Farbeindruck einer Lichtquelle quantitativ zu bestimmen.

Material 1

Entity-Relationship-Modell



Hinweise: Die Herkunft eines Lebewesens speichert sein Ursprungsgebiet (z. B. Südamerika, Kamerun). Die Wuchshöhe der Pflanzenarten und die Länge der Fischarten wird in Zentimeter angegeben. Der Lichtbedarf von Pflanzenarten ist gering, mittel oder hoch. Die Gruppengröße bei Fischen erfasst die optimale Anzahl von Individuen einer Art in einem Aquarium.
Das Lieferdatum wird erst nach erfolgter Lieferung der Bestellung eingetragen.

Material 2

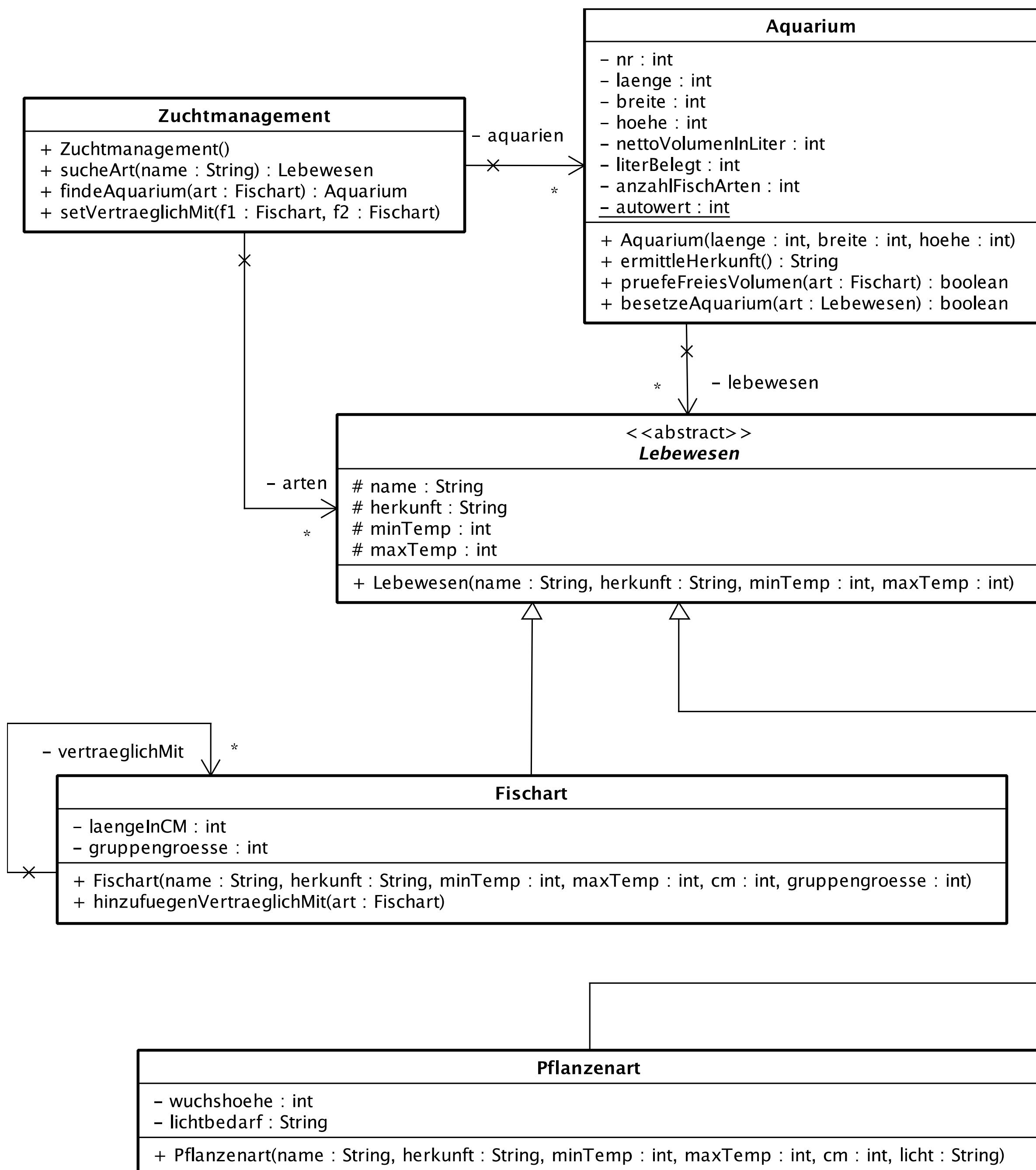
Tabelle Versanddaten (Ausschnitt)

BestNr	Empfänger Lieferadresse	VersandID	ArtNr	An-zahl	Verpackungsmaterial	Versand-datum	Versand-kosten	Transport durch	Liefer-datum
12345	Till Roth Eselsberg 5 89075 Ulm 0173 5566566	541278	7701 7712 5043	15 10 10	2 Transportbeutel 2 Liter 1 Transportbeutel 3 Liter 1 Styroporbox 20 Liter	13.04.22	34,95	Tierexpress 55128 Mainz www.tirex.eu	14.04.22
12346	Ute Schön Kolpingweg 17 32427 Minden 0151 7660033	548877	1981 1238	1 1	2 Transportbeutel 5 Liter 1 Styroporbox 15 Liter	20.04.22	19,95	TuP Versand 53119 Bonn www.itup.eu	22.04.22
		548878	451 971	1 1	Karton 20 Liter	20.04.22	16,49	DHL 53113 Bonn www.dhl.de	23.04.22
12347	Karl Leber Blaubach 11 50676 Köln 0221 976556	548879	2122 4321	2 2	2 Transportbeutel 3 Liter 1 Styroporbox 10 Liter	20.04.22	25,50	TuP Versand 53119 Bonn www.itup.eu	21.04.22
12348	Till Roth Memelstr. 23 89075 Ulm 0173 5566566	548880	7701 7712	10 10	2 Transportbeutel 3 Liter 1 Styroporbox 10 Liter	27.04.22	29,95	Tierexpress 55128 Mainz www.tirex.eu	28.04.22
12349	Jana Lauer Hauptstr. 204 61209 Echzell	548881	123 135	1 1	Karton 15 Liter	27.04.22	5,99	DHL 53113 Bonn www.dhl.de	
...	...		...	...	...	...	...	...	...

Hinweise: Der Versand lebender Artikel (vierstellige Artikelnummern) erfordert entsprechendes Verpackungsmaterial und wird über Firmen abgewickelt, die auf Tiertransporte spezialisiert sind. Zubehör (dreistellige Artikelnummern) wird immer über DHL getrennt versendet. Ein Kunde kann mehrere Lieferadressen haben.

Material 3

UML-Klassendiagramm Zuchtmanagement



Hinweise: Die Einheit aller Längenmaße ist Zentimeter. Auf alle Attribute kann mittels get-Methoden zugegriffen werden.

Material 4

Beispiel-Besatz von Aquarien

Aquarium Nr. 13
Länge 100 cm, Breite 40 cm, Höhe 40 cm
Volumen netto 140 Liter (brutto 160 Liter)

Pflanzenarten				
Name	**Herkunft**	**Wassertemperatur**	**max. Wuchshöhe**	**Lichtbedarf**
Amazonas Schwertpflanze	Südamerika	22 – 28º C	70 cm	mittel
Zwergpfeilkraut	Südamerika	20 – 28º C	10 cm	mittel
Fischarten				
Name	**Herkunft**	**Wassertemperatur**	**Gruppengröße**	**max. Länge**
Blauer Neon	Südamerika	24 – 32º C	15	3 cm
Panda-Panzerwels	Südamerika	23 – 26º C	5	5 cm

Aquarium Nr. 18
Länge 160 cm, Breite 60 cm, Höhe 60 cm
Volumen netto 504 Liter (brutto 576 Liter)

Pflanzenarten				
Name	**Herkunft**	**Wassertemperatur**	**max. Wuchshöhe**	**Lichtbedarf**
Amazonas Schwertpflanze	Südamerika	22 – 28º C	70 cm	mittel
Fischarten				
Name	**Herkunft**	**Wassertemperatur**	**Gruppengröße**	**max. Länge**
Diskus Red spotted green	Südamerika	28 – 30º C	8	23 cm

Hinweise: Das Brutto-Volumen eines Aquariums in Litern berechnet sich aus Länge*Breite* Höhe/1000. Das Netto-Volumen (Attribut `nettoVolumenInLiter`) ist durch Substrat (Kies und Sand) um pauschal 12,5% geringer als das Brutto-Volumen. Das jeweilige Netto-Mindestvolumen für die Fische eines Aquariums berechnet sich nach der Faustregel: Pro Zentimeter Fischlänge (Attribut `laengeInCM`) sind 2 Liter Wasser erforderlich.
Beispiel:
15 Tiere „Blauer Neon" je 3cm*2 Liter = 90 Liter,
5 Tiere „Panda-Panzerwels" je 5 cm*2 Liter = 50 Liter.
Die Summe von 140 Liter beider Volumenergebnisse wird in dem Attribut `literBelegt` gespeichert.
Das Netto-Volumen des Aquariums Nr. 13 ist also mit 140 Litern ausreichend. Der Besatz mit Pflanzen ist bei dieser Berechnung nicht zu berücksichtigen.
Eine Fischart wird in einem Aquarium in einer festgelegten Gruppengröße gehalten (Attribut `gruppengroesse`).
In einem Aquarium werden nur Arten gehalten, die zueinander verträglich sind.

Material 5

Klassendokumentationen

Klasse List

```
List<T>
-----------------------------
+ List<T>()
+ add(obj : T)
+ add(index : int, obj : T)
+ contains(obj : T) : boolean
+ get(index : int) : T
+ remove(index : int) : T
+ remove(obj : T) : boolean
+ size() : int
```

List<T>()
erzeugt eine generische Liste mit Elementen des Typs T.

add(obj: T)
hängt das Objekt obj vom Typ T am Ende der Liste an.

add(index: int, obj: T)
fügt das Objekt obj vom Typ T an der Position index in die Liste ein.

contains(obj: T): boolean
liefert true, wenn das Objekt obj in der Liste enthalten ist, ansonsten false.

get(index: int): T
liefert das Listenelement an der Position index zurück bzw. null, falls index negativ oder größer gleich der Anzahl der momentan enthaltenen Elemente ist.

remove(index: int): T
entfernt das Objekt vom Typ T an der Position index aus der Liste und gibt es zurück.

remove(obj: T): boolean
entfernt das Objekt obj aus der Liste. Falls obj mehrmals in der Liste enthalten ist, wird nur das erste Vorkommen entfernt. Der Rückgabewert ist true, falls das Objekt gefunden und entfernt wurde, sonst false.

size(): int
liefert die Anzahl der Elemente in der Liste zurück.

Klasse VUSB

```
VUSB
-----------------------------
+ VUSB(device : int)
+ writeDigital(wert : int)
+ readDigital() : int
+ readAnalog1() : int
+ readAnalog2() : int
+ close() : void
```

Die Klasse VUSB kapselt ein USB-Interface der Firma Velleman. Es ist möglich, mehrere Interfaces anzusteuern.

VUSB(device: int)
erzeugt ein VUSB-Objekt mit dem übergebenen Device und öffnet die Port-Treiber.

writeDigital(wert: int)
schreibt den Wert wert an die digitalen Ausgänge D0 bis D7. Der Ausgang mit der Nummer n hat die Wertigkeit 2^n.

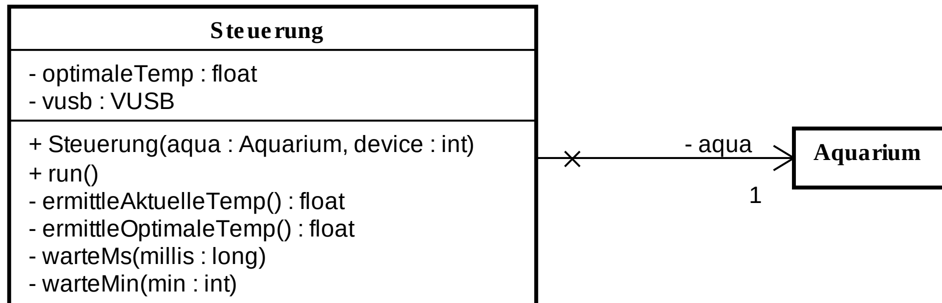
readDigital(): int
liest die digitale Eingabe D0 bis D4 aus und gibt eine ganze Zahl zurück. Die Wertigkeit des Eingangs n ist 2^n.

readAnalog1(): int bzw. readAnalog2(): int
liest vom ersten bzw. zweiten analogen Eingang ein und gibt den Wert (Wertebereich 0 bis 1023) zurück.

close()
schließt die Port-Treiber.

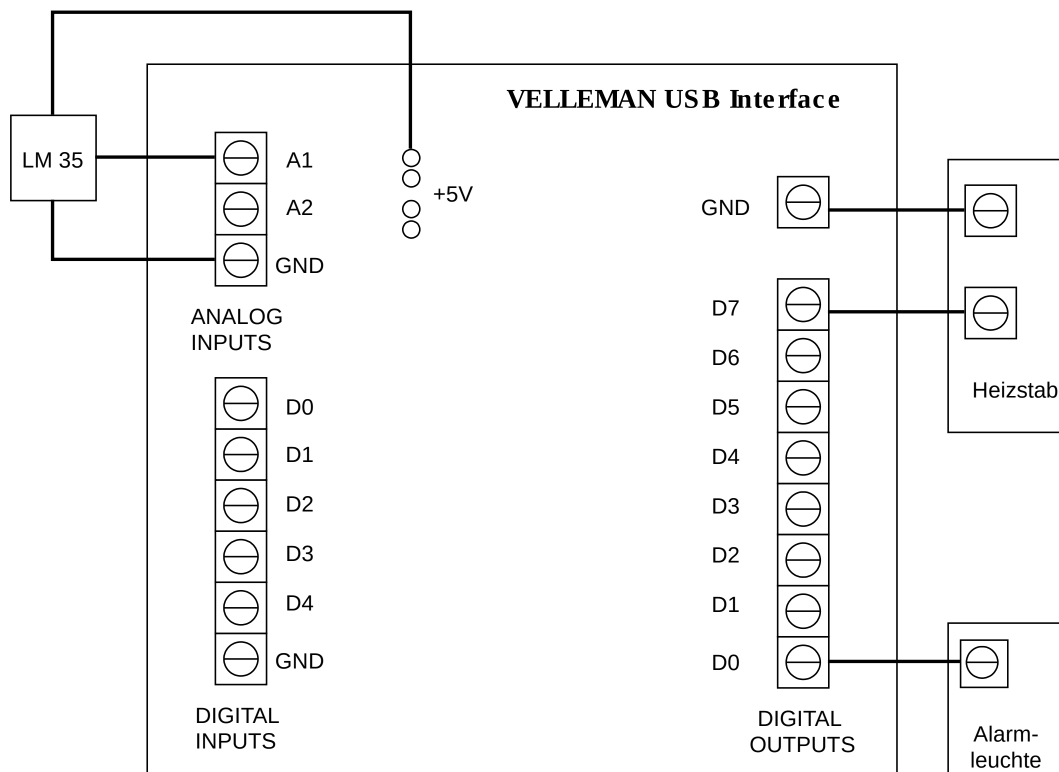
Material 6

UML-Klassendiagramm Aquariumssteuerung



Material 7

Belegung der Ports (schematisch)



Hinweise: Der Temperatursensor LM 35 hat eine Empfindlichkeit von $10\,mV/°C$ und einen messbaren Temperaturbereich von 0 bis 100° Celsius. Die Formel zur Berechnung der Temperatur in Abhängigkeit vom gemessenen Wert am analogen Eingang lautet:

$$Temperatur\ [°C] = \frac{5.0 * 100.0 * analogWert}{1024.0}$$